



# UNITED STATES PATENT AND TRADEMARK OFFICE

50

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/973,748	10/10/2001	Michael Dean Dallin	BLD920010017US1	7134
23550	7590	06/21/2005	EXAMINER	
HOFFMAN WARNICK & D'ALESSANDRO, LLC 3 E-COMM SQUARE ALBANY, NY 12207			MASKULINSKI, MICHAEL C	
			ART UNIT	PAPER NUMBER
			2113	

DATE MAILED: 06/21/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

## Office Action Summary

Application No.

09/973,748

Applicant(s)

DALLIN, MICHAEL DEAN

Examiner

Michael C. Maskulinski

Art Unit

2113

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 19 May 2005.
- 2a) ☐ This action is FINAL. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☐ Claim(s) 1-29 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-4, 6-13, 15-17, 19-24 and 26-29 is/are rejected.
- 7) ☒ Claim(s) 5, 14, 18 and 25 is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 10 October 2001 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
  - ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- \* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: \_\_\_\_\_

**Non-Final Office Action**

***Claim Rejections - 35 USC § 102***

1. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.
2. Claims 1-4, 8, 10-13, 15-17, 19, 20, 23, 24, 26, and 27 are rejected under 35 U.S.C. 102(b) as being anticipated by Smith, Jr., U.S. Patent 5,754,755.

Referring to claims 1 and 10:

a. In column 2, lines 28-30, Smith, Jr. discloses that an output file generator receives an indication of a test template file that has test instructions that contain placeholders (providing a predefined template).

b. In column 2, lines 32-34, Smith, Jr. discloses that the output file generator receives an ordered list of customizing files that have application-specific placeholder values (providing a table having test data for the software product).

Although, Smith Jr. doesn't explicitly disclose a table that is a column-row matrix, an ordered list is inherently a table that is a column-row matrix since it is possible to have a table that is  $1 \times n$  or  $n \times 1$  where  $n$  can be any positive integer.

c. In column 2, lines 34-42, Smith, Jr. discloses that if a test instruction contains a placeholder, then the output file generator searches the customizing files according to the ordered list for a first placeholder value for the placeholder of the test instruction. When such a placeholder value is found, the output file generator inserts the placeholder value into the test instruction. The output file generator then stores the test instruction into the application-specific test script

(running a test generation system with the template to process the test data and to automatically generate a test script file).

d. In column 2, lines 42-44, Smith, Jr. discloses that after the generating is complete, this application-specific test script can then be used to test the application (running the software product while using the generated test script file to test the software product).

Referring to claim 2, in column 2, lines 34-42, Smith, Jr. discloses that if a test instruction contains a placeholder, then the output file generator searches the customizing files according to the ordered list for a first placeholder value for the placeholder of the test instruction. When such a placeholder value is found, the output file generator inserts the placeholder value into the test instruction. The output file generator then stores the test instruction into the application-specific test script (the test script file contains at least one test case generated based upon the test data and the template).

Referring to claim 3, in column 4, lines 43-45, Smith, Jr. discloses that the test template file contains a file type indicator, common macro instructions (a type template having at least one macro), control commands, and placeholders (wherein the type template provides a key for the test generation system to generate the at least one test case based upon the test data).

Referring to claims 4, 13, 20, and 27, in column 4, lines 43-45, Smith, Jr. discloses that the test template file contains a file type indicator, common macro, control

Art Unit: 2113

commands, and placeholders (an output template, wherein the output template has at least one macro and dictates a format of the generated test script file).

Referring to claim 8, in Figure 1, Smith, Jr. discloses that the test generation system and the software product reside on different systems.

Referring to claims 11 and 26, in column 4, lines 43-45, Smith, Jr. discloses that the test template file contains a file type indicator, common macro instructions, control commands, and placeholders (a type template that provides a key for the test generation system to generate the at least one test case based upon the test data).

Referring to claim 12, in column 4, lines 43-45, Smith, Jr. discloses that the test template file contains a file type indicator, common macro instructions (the type template includes at least one macro), control commands, and placeholders.

Referring to claim 15, in column 2, lines 34-42, Smith, Jr. discloses that if a test instruction contains a placeholder, then the output file generator searches the customizing files according to the ordered list for a first placeholder value for the placeholder of the test instruction. When such a placeholder value is found, the output file generator inserts the placeholder value into the test instruction. The output file generator then stores the test instruction into the application-specific test script (the test script file includes at least one test case automatically generated by the test generation system based upon the test data).

Referring to claim 16:

- a. In column 2, lines 28-30, Smith, Jr. discloses that an output file generator receives an indication of a test template file that has test instructions that contain placeholders (entering a predefined type template into a test generation system).
- b. In column 4, lines 56-65, Smith, Jr. discloses a template with control commands (entering a predefined output template into the test generation system).
- c. In column 2, lines 32-34, Smith, Jr. discloses that the output file generator receives an ordered list of customizing files that have application-specific placeholder values (entering a table having test data for the software product into the test generation system).
- d. In column 2, lines 34-42, Smith, Jr. discloses that if a test instruction contains a placeholder, then the output file generator searches the customizing files according to the ordered list for a first placeholder value for the placeholder of the test instruction. When such a placeholder value is found, the output file generator inserts the placeholder value into the test instruction. The output file generator then stores the test instruction into the application-specific test script (running the test generation system with the type template and the output template on a first computer system to process the test data to automatically generate an executable test script file).
- e. In column 2, lines 42-44 and in Figure 1, Smith, Jr. discloses that after the generating is complete, this application-specific test script can then be used to test the application (running the software product on a second computer while

using an automation tool to run the executable test script file to test the software product).

Referring to claim 17:

a. In column 2, lines 28-30, Smith, Jr. discloses that an output file generator receives an indication of a test template file that has test instructions that contain placeholders. In column 4, lines 56-65, Smith, Jr. discloses a template with control commands. In column 2, lines 32-34, Smith, Jr. discloses that the output file generator receives an ordered list of customizing files that have application-specific placeholder values (an input system for receiving a type template, an output template, and a table having test data for the software product).

b. In column 2, lines 34-42, Smith, Jr. discloses that if a test instruction contains a placeholder, then the output file generator searches the customizing files according to the ordered list for a first placeholder value for the placeholder of the test instruction. When such a placeholder value is found, the output file generator inserts the placeholder value into the test instruction. The output file generator then stores the test instruction into the application-specific test script. In column 2, lines 42-44 and in Figure 1, Smith, Jr. discloses that after the generating is complete, this application-specific test script can then be used to test the application (a process system for processing the test data to automatically generate a test script file based upon the type template and the output template, wherein the generated test script file is used to test the software product).

Referring to claim 19, in column 4, lines 43-45, Smith, Jr. discloses that the test template file contains a file type indicator, common macro instructions, control commands, and placeholders (the type template provides a key for generating the test script file based upon the test data). Further, in column 3, lines 5-28, Smith, Jr. teaches that the tests script file includes at least one test case.

Referring to claim 23:

- a. In Figure 1, Smith, Jr. discloses a test generation system stored on a first computer system.
- b. In column 2, lines 28-30, Smith, Jr. discloses that an output file generator receives an indication of a test template file that has test instructions that contain placeholders. In column 4, lines 56-65, Smith, Jr. discloses a template with control commands. In column 2, lines 32-34, Smith, Jr. discloses that the output file generator receives an ordered list of customizing files that have application-specific placeholder values (means for receiving a type template, an output template, and a table having test data for the software product).
- c. In column 2, lines 34-42, Smith, Jr. discloses that if a test instruction contains a placeholder, then the output file generator searches the customizing files according to the ordered list for a first placeholder value for the placeholder of the test instruction. When such a placeholder value is found, the output file generator inserts the placeholder value into the test instruction. The output file generator then stores the test instruction into the application-specific test script. In column 2, lines 42-44 and in Figure 1, Smith, Jr. discloses that after the

generating is complete, this application-specific test script can then be used to test the application (means for processing the test data, based upon the type template and the output template, to automatically generate an executable test script file having at least one test case).

f. In Figure 1, Smith, Jr. discloses a means for outputting the executable test script file.

g. In column 2, lines 42-44 and in Figure 1, Smith, Jr. discloses that after the generating is complete, this application-specific test script can then be used to test the application (an automation toll for running the executable test script file to test the software product, wherein the software product is stored on a second computer system).

Referring to claim 24:

a. In column 4, lines 40-41, Smith, Jr. discloses that the output file generator may be stored as instructions on a computer-readable medium such as a disk (a program product, stored on a recordable medium, for testing a software product).

b. In column 2, lines 28-30, Smith, Jr. discloses that an output file generator receives an indication of a test template file that has test instructions that contain placeholders. In column 4, lines 56-65, Smith, Jr. discloses a template with control commands. In column 2, lines 32-34, Smith, Jr. discloses that the output file generator receives an ordered list of customizing files that have application-specific placeholder values (an input system for receiving a type template, an output template, and a table having test data for the software product).

c. In column 2, lines 34-42, Smith, Jr. discloses that if a test instruction contains a placeholder, then the output file generator searches the customizing files according to the ordered list for a first placeholder value for the placeholder of the test instruction. When such a placeholder value is found, the output file generator inserts the placeholder value into the test instruction. The output file generator then stores the test instruction into the application-specific test script. In column 2, lines 42-44 and in Figure 1, Smith, Jr. discloses that after the generating is complete, this application-specific test script can then be used to test the application (a process system for processing the test data to automatically generate a test script file based upon the type template and the output template, wherein the generated test script file is used to test the software product).

***Claim Rejections - 35 USC § 103***

3. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

4. Claims 6, 7, 9, 21, 22, 28, and 29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Smith, Jr., U.S. Patent 5,754,755.

Referring to claims 6, 21, and 28, in column 2, lines 42-44, Smith, Jr. discloses that the application-specific test script can be used to test the application. However, Smith, Jr. doesn't explicitly disclose automatically running the test script with an automation tool. In column 1, lines 23-25, Smith, Jr. discloses in the Background of the Invention that some conventional testing procedures involve first manually writing a test

Art Unit: 2113

script and then using that test script to automatically test the application program. It would have been obvious to one of ordinary skill at the time of the invention to include the automatic testing of the application program into the system of Smith, Jr. A person of ordinary skill in the art would have been motivated to make the modification *because of the importance of thorough testing and because such testing can be very time-consuming, the application developers have developed extensive automated testing procedures* (see Smith, Jr.: column 1, lines 17-21).

Referring to claims 7, 22, and 29, in column 2, lines 42-44, Smith, Jr. discloses that the application-specific test script can be used to test the application. However, Smith, Jr. doesn't explicitly disclose running the test script manually by a user. The Examiner takes Official Notice that in the art of software testing it is well known to use a stepping procedure to step through a program in order to debug a program. A user does this manually. It would have been obvious to one of ordinary skill at the time of the invention to include the manual testing of the application program into the system of Smith, Jr. A person of ordinary skill in the art would have been motivated to make the modification because manually stepping through a program's instructions allows a user to find specific lines of code that caused the problem.

Referring to claim 9, in column 4, lines 40-41, Smith, Jr. discloses that the output file generator may be stored as instructions on a computer-readable medium such as a disk. However, Smith, Jr. discloses that the test generation system and the software product reside on the same computer system. It would have been obvious to one of ordinary skill at the time of the invention to use the disk to install the output file

Art Unit: 2113

generator instructions onto the same computer system that the software product resides on. A person of ordinary skill in the art would have been motivated to make the modification because the whole point of storing instructions on a disk is so that it can be run and stored anywhere.

### ***Allowable Subject Matter***

5. Claims 5, 14, 18, and 25 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

### ***Response to Arguments***

6. Applicant's arguments filed April 18, 2005 have been fully considered but they are not persuasive.

7. On pages 9-10, under section A. Rejection of Claims 1-5, 8, 10-20, and 23-27 under 35 U.S.C. 102(b), the Applicant argues, "Applicant submits that Smith, Jr. fails to teach providing a table having test data for the software product, the table being a column-row matrix of test data. Instead, the output file generator in Smith, Jr. '...receives an ordered list of customizing files that have application-specific placeholder values.' Col. 2, lines 32-34. However, the customizing files of Smith, Jr. are specified as being an ordered list and not a table. Furthermore, the ordered list of Smith, Jr. is only specified as having application-specific placeholder values and not as being a column-row matrix of 'test data.' Thus, nowhere does Smith, Jr. teach that its ordered list of customizing files is a table or that the table is a column-row matrix of test data."

Art Unit: 2113

The Examiner respectfully disagrees. In column 4, lines 63-65, Smith, Jr. discloses that the output file generator replaces a placeholder by searching for a placeholder value with that placeholder in the ordered list of customizing files. The ordered list of customizing files is a table. It is a list that is searched by an index value and then returns a corresponding value or function. A table works the same way when you search it, for example, a person would go down a column looking for a value and then look across the row to return the corresponding value. The Applicant is also reminded that a table can be a single entry, a  $1 \times n$  matrix, or even an  $n \times 1$  matrix where  $n$  is any positive integer.

8. Applicant's arguments, with respect to claims 5, 14, 18, and 25 have been fully considered and are persuasive. The rejection of claims 5, 14, 18, and 25 has been withdrawn.

9. On pages 12-13 under section B. Rejection of Claims 6, 7, 9, 21, 22, 28, and 29 under 35 U.S.C. 103(a), the Applicant argues, "Smith, Jr. teaches away from an executable file, in that the test script generated by Smith, Jr., instead of automatically running with an automation tool, appears to be written in a macro language that is capable of running independently of an automation tool. Col. 2, lines 8-22; FIG. 2." The Examiner respectfully disagrees. The section cited by the Applicant discloses that the application program executes the macro program as a test script. This is consistent with the other section of the Background cited above in the rejection. This is clearly an executable file that is automatically run by an automation tool. If the Applicant believes

Art Unit: 2113

otherwise, then the Examiner requests that the Applicant show how it isn't an executable file that is automatically run by an automation tool.

10. On page 11, under section B. Rejection of Claims 6, 7, 9, 21, 22, 28, and 29 under 35 U.S.C. 103(a), the Applicant argues, "this factual assertion is unsubstantiated and is not properly based upon common knowledge. Furthermore, the test script generated by Smith, Jr., instead of being a documentation file that is manually run by a user, appears to be written in a macro language that is capable of running independently of a user." The Examiner respectfully disagrees for at least the reasons given in the Final Office Action, mailed February 18, 2005. Further, in that same Office Action the Examiner provided U.S. Patent 4,177,520 and U.S. Patent 5,121,472 to show single-stepping and its application in debugging programs to show common knowledge. Until, the Applicant acknowledges this and **properly** replies to the Examiner's arguments, the argument will be considered moot.

### ***Conclusion***

11. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. The cited prior art shows additional means of testing software applications.

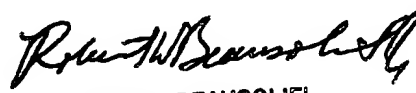
Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael C. Maskulinski whose telephone number is (571) 272-3649. The examiner can normally be reached on Monday-Friday 9:30-6:00.

Art Unit: 2113

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Robert W. Beausoliel can be reached on (571) 272-3645. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

MM

  
ROBERT BEAUSOLIEL  
SUPERVISORY PATENT EXAMINER  
TECHNOLOGY CENTER 2100